mathematics

Article

Point-Wise Ribosome Translation Speed Prediction with Recurrent Neural Networks

Pietro Bongini [1,*], Niccolò Pancino [1], Veronica Lachi [1], Caterina Graziani [1], Giorgia Giacomini [2], Paolo Andreini [1] and Monica Bianchini [1]

[1] Department of Information Engineering and Mathematics, University of Siena, Via Roma 56, 53100 Siena, Italy; niccolo.pancino@unisi.it (N.P.); veronica.lachi@student.unisi.it (V.L.); caterina.graziani@student.unisi.it (C.G.); paolo.andreini@unisi.it (P.A.); monica.bianchini@unisi.it (M.B.)

[2] IRCCS Ospedale San Raffaele, Via Olgettina 60, 20132 Milano, Italy; giacomini.giorgia@hsr.it

* Correspondence: pietro.bongini@unisi.it

Abstract: *Escherichia coli* is a benchmark organism, which has been deeply studied by the scientific community for decades, obtaining a vast amount of metabolic and genetic data. Among these data, estimates of the translation speed of ribosomes over their genome are available. These estimates are based on Ribo-Seq profiles, where the abundance of a particular fragment of mRNA in a profile indicates that it was sampled many times inside a cell. Various measurements of Ribo-Seq profiles are available for *Escherichia coli*, yet they do not always show a high degree of correspondence, which means that they can vary significantly in different experimental setups, being characterized by poor reproducibility. Indeed, within Ribo-Seq profiles, the translation speed for some sequences is easier to estimate, while for others, an uneven distribution of consensus among the different estimates is evidenced. Our goal is to develop an artificial intelligence method that can be trained on a small pool of highly reproducible sequences to establish their translation rate, which can then be exploited to calculate a more reliable estimate of the translation speed on the rest of the genome.

Keywords: translation speed; ribosomes; *Escherichia coli*; Ribo-Seq profiles; neural networks; recurrent neural networks; translation speed prediction

MSC: 68T05

Citation: Bongini, P.; Pancino, N.; Lachi, V.; Graziani, C.; Giacomini, G.; Andreini, P.; Bianchini, M. Point-Wise Ribosome Translation Speed Prediction with Recurrent Neural Networks. *Mathematics* **2024**, *12*, 465. https://doi.org/10.3390/math12030465

Academic Editor: Vince Grolmusz

Received: 19 December 2023
Revised: 22 January 2024
Accepted: 29 January 2024
Published: 31 January 2024

1. Introduction

Recognition of the importance of RNA in biological processes is increasing exponentially. Once considered a simple messenger of information between DNA and proteins, in recent decades, RNA has proven to be essential for regulation of many cellular processes, such as the modulation of gene expression, chromatin structure, and various aspects of genome stability, directly implicated in important pathologies such as tumors and aging. However, the direct contribution of RNA to a specific biological mechanism is often difficult to analyze due to the intrinsic fragility of the molecule. Since RNA represents an intermediate language between DNA and proteins, an accurate prediction of RNA properties is important to understand gene regulation and expression of protein products. In fact, it is a recent discovery that many RNAs also have catalytic properties; they are called ribozymes, and they are involved in the splicing of tRNA molecules, in the activity of ribosomes, in the eukaryotic hnRNA processing, etc. Moreover, RNA acts as a structural scaffold for the DNA, RNA, and polypeptide reactions. Finally, because some viruses, such as HIV, are encoded in the form of RNA, understanding RNA characteristics can support the process of discovery and testing of pharmacological agents against such pathogens.

Therefore, the molecular process of translating mRNA into proteins is a cornerstone of cellular biology, representing a critical intersection between the genetic code and functional biomolecules. The translation is executed by ribosomes, which are sophisticated molecular complexes composed of RNA and protein components. These complexes function as the sites of protein synthesis, interpreting the genetic information encoded in mRNA sequences and assembling corresponding amino acids into polypeptide chains, which subsequently fold into functional proteins essential for cell life.

Translation is not merely a mechanical process but is intricately regulated, playing a pivotal role in the control of gene expression. This regulation is essential for maintaining cellular homeostasis, enabling cells to adapt protein production to suit specific tissue requirements and to respond to a wide variety of internal and external stimuli. The fidelity and efficiency of translation are critical, and disruptions in these processes are frequently implicated in disease mechanisms [1], highlighting the importance of understanding translation at a molecular level. A central aspect of translation that has garnered significant interest is the speed of ribosomal protein synthesis. This rate is not constant but varies based on several factors. The interaction dynamics between the ribosome, mRNA bases, and the amino acids incorporated into the growing polypeptide chain are crucial determinants of this rate. The chemical composition of the mRNA sequence and the encoded amino acid sequence are known to significantly influence translation speed. For instance, the presence of certain amino acids, such as those that are positively charged, can have a pronounced effect on the rate of translation [2].

The complexity of translation extends beyond the ribosome-mRNA interactions. The ribosome itself is a dynamic entity, capable of various interactions with the mRNA molecule, the emerging peptide chain, and external molecular factors [3]. These interactions are not merely mechanical but are intricately regulated, contributing to the efficiency of protein synthesis. Recognizing these interactions is crucial for a comprehensive understanding of translation and its role in cellular function and pathology.

In order to understand if biological signals exist to suggest how ribosomes move on the mRNA strand, we decided to process the mRNA sequences with machine learning models. The objective is to predict the translation speed of each nucleotide, codon, or amino acid inside a sequence. The models are trained, validated, and tested on a dataset of *E. coli* Open Reading Frames (ORFs), obtained from a consensus pool of nine source datasets. Once trained and successfully validated, the model can then be exploited to predict the translation speed of all the *E. coli* ORFs for which the consensus threshold was not reached, thus marking the uncertainty in determining the translation speed with traditional methods. On the one hand, this method can help build models for reliable predictions of mRNA translation speed, reducing the future need for more costly Ribo-Seq experiments. On the other hand, ablation studies and attention mechanisms can help explain the models' decisions, thus identifying the factors that determine the speed in nature and their importance.

The main contributions of this work are the following:

- We have developed, optimized, and rigorously compared four advanced machine learning models, each designed to process and interpret *E. coli* mRNA sequences;
- We have explored and evaluated four distinct encoding strategies to determine the most effective method for representing these sequences, ensuring that our models receive data in a format that maximizes their predictive capabilities;
- We conducted an in-depth analysis of the impact of context length on model performance; this analysis aims to identify the optimal context length that provides the most complete and informative data to predict translation speed;
- We have implemented and analyzed an attention mechanism within our models. This mechanism is designed to identify and quantify the parts of the mRNA sequence that are most influential in determining translation speed, offering insights into the molecular determinants of this key biological process.

The rest of the paper is organized as follows. In Section 2, we introduce the dataset and the methodology. In particular, in Section 2.1, we briefly introduce the procedure used to build the dataset and the encoding techniques; in Section 2.2, we define the problem and how we propose to solve it; in Section 2.3, we describe the models employed and how they can help solve our task; in Section 2.4, we summarize the experimental setup. In Section 3, we describe the results of the experiments and introduce their significance. Finally, in Section 4, we discuss the results and give conclusions, identifying interesting directions for future research.

2. Materials and Methods

2.1. Dataset

Our dataset was obtained as a consensus pool of sequences from 9 different Ribo-Seq profile sources. The source datasets were collected from the GEO repository [4]. Each of them was built as a result of Ribo-Seq experiments carried out by culturing wild-type *Escherichia coli* in different setups, and each was obtained from a different GEO sample. See Table 1 for reference. The consensus pool was obtained using the algorithm described in [5] and consisted of 49 sequences. The other *E. coli* ORFs did not show a sufficient consensus between the sources to be used as supervision for our models. The objective was to train the model on the 49 reliable ORFs and then exploit the acquired knowledge to make a more reliable estimation of the rest of the ORFs.

Table 1. Reference information for the 9 datasets used to build our pool of consensus sequences.

Dataset ID	GEO Series ID	GEO Sample ID	Ref
Dataset 1	GSE64488	GSM1572266	[6]
Dataset 2	GSE90056	GSM2396722	[7]
Dataset 3	GSE72899	GSM1874188	[8]
Dataset 4	GSE53767	GSM1300279	[9]
Dataset 5	GSE51052	GSM1399615	[10]
Dataset 6	GSE58637	GSM1415871	[11]
Dataset 7	GSE77617	GSM2055244	[12]
Dataset 8	GSE35641	GSM872393	[13]
Dataset 9	GSE88725	GSM2344796	[14]

The dataset made labels available for the 49 ORF sequences on which we had consensus among all the sources. In particular, as described in [5,15], we labeled a data point as "fast" (+1) or "slow" (-1) if 75% of the sources agree, respectively, on a low or high Ribo-Seq profile for that data point. Data points on which there was no agreement were left with a 0 label (which corresponded to a neutral supervision for the machine learning models). All the other *E. coli* ORFs were unlabeled, and our objective was to formulate reliable estimations of the translation speed on these latter sequences, using our predictor trained and tested on the 49 labeled sequences.

Our 49 sequences have variable length, with the longest spanning 4461 nucleotides and the shortest just 150. The average length is about 1934.6 nucleotides. They amount to a total of 94,794 nucleotides, encoding 31,598 codons that translate to 31,549 amino acids and 49 stop codons.

Four different encodings of this dataset were used. To assess the informativity of each encoding and select the best one, we used:

- A nucleotide encoding (N), in which the mRNA nucleotides are treated separately, using a one-hot encoding of length 4;
- A codon encoding (C), in which codons are encoded by concatenating the 3 one-hot encodings of their components, for a total length of 12;
- A spread codon encoding (S), in which codons are encoded one by one with a one-hot encoding of length 64;
- An amino acid encoding (A), in which codons are encoded using the one-hot encoding of the amino acids they code for, plus the stop codon, therefore using a one-hot encoding of length 21.

As a consequence, N encodings have a triple sequence length with respect to all other encodings. Also, context lengths will always be tripled in experiments carried out on N encodings to match the same sequence portion used in the other experiments.

2.2. Point-Wise Speed Prediction with Deep Neural Networks

Being determined by the sequence of mRNA bases, the speed of translation can be predicted with a neural network model capable of processing sequential data. Sequential models such as LSTMs [16] and 1D-CNNs [17] seem particularly fit for this task, as well as more general models that can be adapted to this case, namely GNNs (GNNs can process any graph, and sequences are particular cases of graphs) [18]. The objective is to formulate a point-wise prediction of ribosome translation speed over the input sequence. We exploited the Ribo-Seq profiles of the 49 sequences with high consensus among the sources as our supervisions. These allowed us to train and validate our models before using them to predict the translation speed over the rest of the *E. coli* ORFs. The point-wise prediction was formulated on every element of the sequence: each nucleotide, amino acid, or codon (depending on how the problem is formulated) has its speed value predicted.

Since the models take into account the dependencies between nearby sequence elements, it is important to evaluate the best context width before making the predictions. The context is the window of sequence positions surrounding the position for which we are predicting the speed. It can be as large as the whole sequence or as small as the single sequence position itself. Previous biological studies suggest that the span of the mRNA sequence interacting with the ribosome (and therefore capable of influencing the translation speed) has a length of about ten codons [19] and is slightly unbalanced towards the tail: 4 codons forward, 5 codons backward, with respect to the one being translated.

We carried out a comparison between the three models introduced above and a hybrid version of GNNs, which uses an LSTM to aggregate nodes (which will be referred to as the hybrid model). To build, train, and validate the models, we used the TensorFlow Keras Python framework [20,21]. All the models were trained using the Adam optimizer [22].

2.3. The Models

The Long Short-Term Memory (LSTM [16]) model is a type of recurrent neural network (RNN) designed for processing sequences of data to capture long-term dependencies between elements. The central role of a common LSTM unit is held by the “cell”, a unit acting as a memory, capable of maintaining its state over time. Information can be added to or removed from the cell state in LSTMs and is regulated by three gates, namely the Forget Gate, Input Gate, and Output Gate, to selectively store, discard, and output information, by means of a mechanism based on point-wise multiplications and sigmoid activation functions. Bidirectional LSTMs represent an advancement beyond LSTMs since each training sequence is processed both forward and backward, effectively employing separate LSTM networks. As a result, a Bidirectional-LSTM model possesses comprehensive information about every element in a given sequence, both before and after it.

Convolutional Neural Networks (CNNs [17]) in 1D operate by leveraging convolutional layers to analyze sequential data. These layers use filters that slide along the input sequence, capturing local patterns and hierarchies and enabling automatic feature extraction without manual engineering. They efficiently learn and identify relevant features, making them adaptable to diverse applications. The inclusion of pooling layers aids in downsampling and maintaining essential information while enhancing computational efficiency. In essence, 1D-CNNs work by autonomously recognizing and utilizing meaningful features in one-dimensional sequences, providing a powerful tool for analyzing and extracting patterns from sequential data.

Based on an information diffusion mechanism, GNNs can process graph-structured data [18,21]. Specifically, GNNs create an encoding network, a recurrent neural network that replicates the input graph’s topology. This network comprises two MLP units: one implementing a state transition function for each node and the other the output function (on specific nodes or edges). The GNN employs a message-passing algorithm for exchanging information between nodes and their neighbors, either for a predetermined number of iterations T or until the state computation dynamics converge to a stable equilibrium point at $t \leq T$. The final versions of the node states, $x_n^T, \forall n \in N$, are fed in input to the output network, which approximates a function that can be defined on single nodes, edges, or the whole graph. To produce an output, the GNN replicates the MLP units on each node of the input graph and unfolds itself in time and space, generating a feedforward architecture known as the unfolding network, in which each layer contains copies of all the elements of the encoding network and represents an iteration of the implemented algorithm. Connections between neurons belonging to subsequent layers reproduce exactly those of the encoding network. Through a series of iterations, the information associated with each node can be effectively propagated throughout the entire graph.

In the hybrid model setting, an LSTM model is incorporated within a GNN architecture as a mechanism for aggregating messages exchanged between nodes and their neighborhood to capture complex relationships and dependencies.

2.4. Experimental Setup

To perform the machine learning experiments, the 49 sequences were divided into a training set (42 sequences), a validation set (3 sequences), and a test set (4 sequences). All experiments used the same split of the dataset.

The first experiments were devoted to finding the best model layouts. For every model, multiple configurations were tested using a grid search-like procedure. The objective was to find the best model capable of processing our mRNA sequences and to learn the best hyperparameters for the task. All the models have an early stopping procedure that evaluates the loss function on the validation set to prevent overfitting. When this occurs, the best configuration is restored based on the best validation loss value. In this phase, with respect to the C encoding, we use a context based on the literature [19], which includes 4 codons in the forward context and 5 codons in the backward context, for a total length of 10 codons (including the one being predicted). Appropriate padding was applied to the head and tail of each sequence (4 vectors of zeros before the head, 5 after the tail). The grid search was carried out using the validation set for model evaluation. In particular, we used the hyperparameters reported in Table 2.

After the grid search had produced the best model configurations on all the dataset encodings, we proceeded with a comparison between them in order to determine the best model for our task. In this case, the test set was used for measuring the performance. The best architecture of each model on each of the encodings was used in this set of experiments. Each experiment was repeated five times, and we calculated the average value and standard deviation for each metric.

Once the best model had been selected, we tuned the context length to the task at hand. We varied the context radius from 1 to 15 codons in order to encompass a growing number of sequence positions in our context. A context radius of X corresponds to a total length of 2X + 1 because the context unfolds in both directions, and we also consider the central element for which the predictions are formulated. The padding was also adjusted accordingly for each experiment. On the one hand, reducing the context too much could bring a loss of valuable information, as close neighbors likely have a certain influence on the speed at which our central position is translated. On the other hand, spreading the context too much could lead to the introduction of data that do not convey useful information because far positions have very little to no influence on the translation speed of our central position. The context variation experiments were carried out with the LSTM only, as this was identified as the best model in the previous set of experiments. Again, each experiment was repeated five times, measuring the average value and standard deviation of the metrics over the five runs.

Table 2. Best model configurations obtained with the grid search. Hyperparameters are explained in the following. E: epochs (All), R: learning rate (All), L1: units in LSTM layer 1 (Hybrid and LSTM), L2: units in LSTM layer 2 (Hybrid), D: units in dense layer (Hybrid, LSTM, and CNN), S1: units in layer 1 of state updating network (GNN), S2: units in layer 2 of state updating network (GNN), O: units in output network layer (GNN), St 1: Stride of layer 1 (CNN), St 2: Stride of layer 2 (CNN), K 1: Kernel size in layer 1 (CNN), K 2: Kernel size in layer 2 (CNN), F 1: Number of filters in layer 1 (CNN), F 2: Number of filters in layer 2 (CNN). The #P row gives the total number of parameters of each model.

Model	**LSTM**				**Hybrid**				**GNN**				**CNN**			
Encoding	**C**	**N**	**A**	**S**	**C**	**N**	**A**	**S**	**C**	**N**	**A**	**S**	**C**	**N**	**A**	**S**
E	500	800	800	800	500	800	800	800	500	800	800	800	500	800	800	800
R	1×10^{-3}	1×10^{-3}	1×10^{-3}	1×10^{-3}	1×10^{-3}	5×10^{-3}	1×10^{-3}	1×10^{-3}	1×10^{-3}	1×10^{-4}	1×10^{-3}	1×10^{-3}	1×10^{-3}	5×10^{-3}	1×10^{-3}	1×10^{-3}
L1	12	12	12	7	7	8	5	3	-				-			
L2	-				7	8	5	3	-				-			
D	50	30	30	30	30	50	36	18	-				32	32	30	16
S1	-				-				35	38	33	31	-			
S2	-				-				35	38	33	31	-			
O	-				-				30	30	30	30	-			
St 1	-				-				-				1	1	1	1
St 2	-				-				-				1	1	1	1
K 1	-				-				-				5	7	7	3
K 2	-				-				-				3	5	5	3
F 1	-				-				-				16	16	16	16
F 2	-				-				-				8	8	8	8
#P	3262	3014	3826	4444	3162	3270	3436	4630	3275	3054	3619	4027	3002	3002	3636	4170

3. Results

After having determined the best-performing configurations with the grid search, which resulted in the architectures described in Table 2, a comparison was carried out with the objective of determining the best model. The results of these experiments on the four different dataset encodings are summarized in Table 3.

Table 3. Comparison of model performance on the four different dataset encodings. Precision, recall, accuracy, and F1 Score were measured on five experiment repetitions on the same dataset split for each model. The average value and standard deviation are reported. The letter following each model accounts for the encoding: C stands for codon, N for nucleotide, A for amino acid, and S for spread codon. Please refer to Section 2.1 for an explanation of the encodings and to Section 2.4 for the model configurations.

Model	Precision	Recall	Accuracy	F1 Score
LSTM-C	82.87% ± 0.15%	74.07% ± 0.16%	78.14% ± 0.11%	78.22% ± 0.11%
Hybrid-C	84.57% ± 1.50%	72.53% ± 1.04%	78.27% ± 0.70%	78.21% ± 0.91%
GNN-C	71.04% ± 3.55%	79.75% ± 2.94%	75.15% ± 3.14%	75.12% ± 2.90%
CNN-C	82.75% ± 2.40%	72.01% ± 1.81%	77.17% ± 1.26%	76.98% ± 1.15%
LSTM-N	75.66% ± 0.38%	63.69% ± 0.59%	71.47% ± 0.14%	69.16% ± 0.25%
Hybrid-N	74.09% ± 0.53%	62.87% ± 0.53%	70.31% ± 0.38%	68.02% ± 0.42%
GNN-N	66.76% ± 1.18%	74.45% ± 3.23%	68.81% ± 1.22%	70.36% ± 1.55%
CNN-N	73.73% ± 2.42%	62.58% ± 1.52%	69.96% ± 1.17%	67.66% ± 0.85%
LSTM-A	79.17% ± 0.38%	76.26% ± 1.11%	76.78% ± 0.29%	77.68% ± 0.45%
Hybrid-A	78.17% ± 0.84%	76.56% ± 0.78%	76.23% ± 0.40%	77.35% ± 0.31%
GNN-A	72.67% ± 0.97%	73.39% ± 1.19%	74.52% ± 0.86%	73.03% ± 0.94%
CNN-A	78.61% ± 0.56%	77.51% ± 0.96%	76.89% ± 0.48%	78.05% ± 0.51%
LSTM-S	83.15% ± 1.77%	73.55% ± 0.60%	78.06% ± 1.00%	78.05% ± 0.79%
Hybrid-S	83.71% ± 0.43%	73.77% ± 0.99%	78.46% ± 0.50%	78.45% ± 0.61%
GNN-S	71.29% ± 2.79%	81.16% ± 4.15%	75.65% ± 1.20%	75.79% ± 0.90%
CNN-S	83.51% ± 0.77%	74.14% ± 0.71%	78.52% ± 0.49%	78.54% ± 0.48%

As shown in Table 3, the models are all capable of learning the task from the available data. General models like GNNs clearly cannot reach the same performance levels as more specialized ones. Even the hybrid GNN with an LSTM aggregation mechanism is outperformed by the 1D-CNN and, more importantly, by the LSTM. Overall, this latter model has the best performance levels and clearly represents the best model for our task. The encodings are also very important in determining the model performance: when using nucleotide encodings, we got an F1 Score just above 70%, while all the other three encodings brought at least one model above 78%. This suggests that what determines the translation speed of a mRNA sequence is mainly the peptide chain it codes for, while the nucleotide sequence itself plays a secondary role. This might be due to drag levels and chemical bonds formed between the peptide chain and the ribosome, while forces between the ribosome and the mRNA sequence could be significantly weaker.

To further explore the correlation between the sequence and its translation speed, we carried out a series of context variation experiments. In this scope, we varied the length of the context on the two encodings, which were associated with the best results in the grid search (S and A), using the best model resulting from the comparison: the LSTM. The architecture has the same configuration described in Table 2. The context variation results are summarized in Figure 1 (for the amino acid encoding) and Figure 2 (for the spread codon encoding).

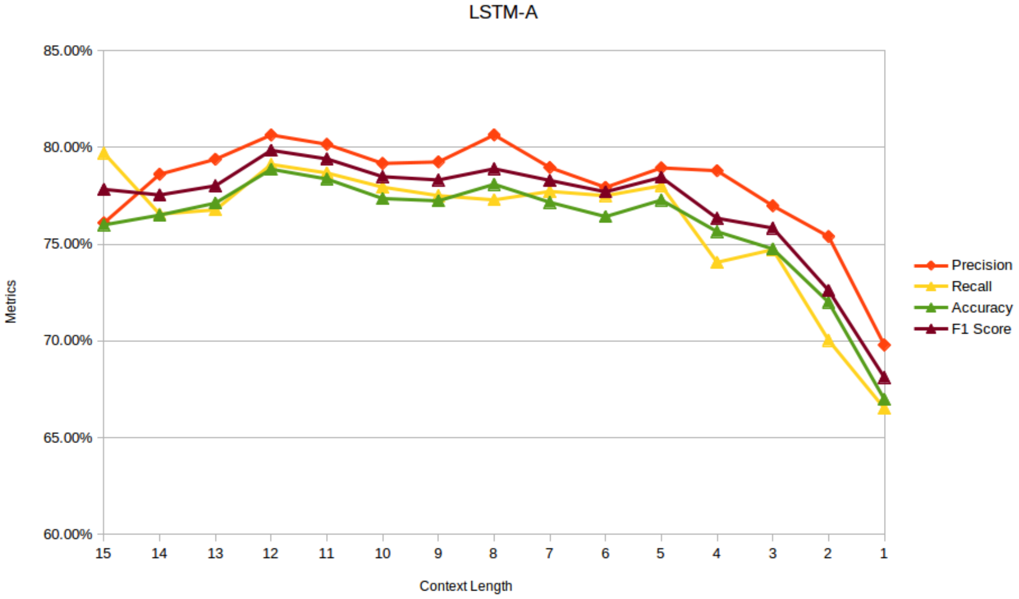

Figure 1. Context variation results with the amino acid encoding.

As we can see in Figure 1, the context that maximizes the performance includes 12 amino acids, corresponding to a total window width of 25 amino acids (12 forward, 12 backward, plus the central one). This suggests that the region of interaction with the translating ribosome spans for a similar length over the sequence. A similar observation can be made for codons: as per Figure 2, the optimal context length is 11 codons for a total window width of 23 codons.

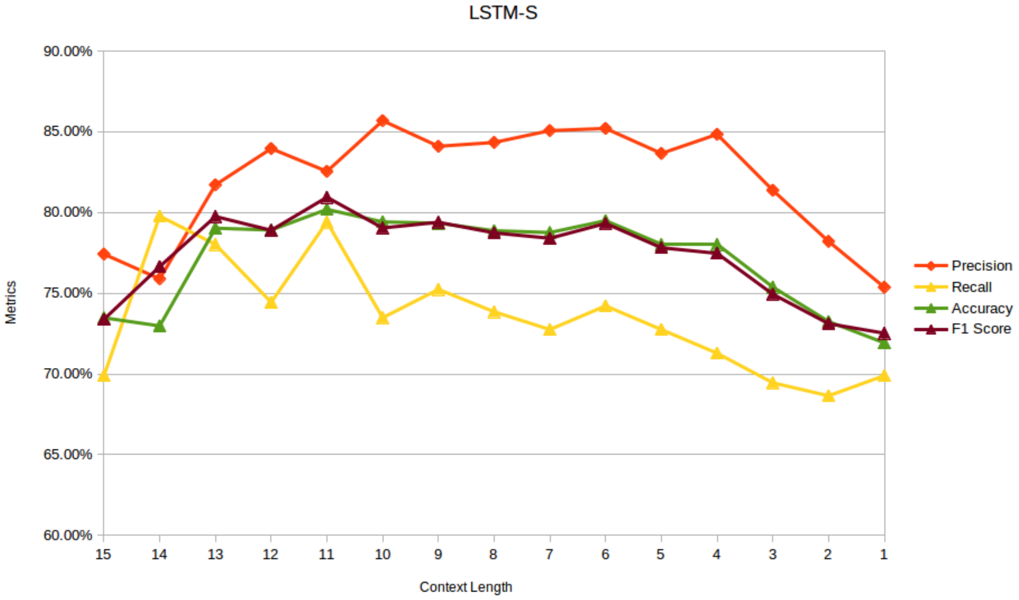

Figure 2. Context variation results with the spread codon encoding.

Moreover, we carried out an additional experiment with the two LSTM configurations resulting from these experiments. We inserted an attention layer in the models in order to measure the importance given by the model itself to each sequence position and each possible amino acid/codon. After training the model with the attention layer, the attention levels can be measured by feeding data to it and recording the response of the attention layer. We realized a heat-map for each input sequence position and then averaged the results over the whole test set, as shown in Figure 3 for amino acids and in Figure 4 for codons.

Interestingly, the position one step forward (1) with respect to the current translating spot (0) is evaluated as the most important in both cases. The first backward position (−1) is also under the focus of the attention layer, with position 0 often coming third in order of attention level. Moreover, in both heat maps, it can be observed that the backward context has a higher level of attention, which also reaches positions further away than the forward context.

In order to finalize the work, we then used the "optimal" models obtained in the previous experiments to actually make predictions on all the other *E. coli* ORFs for which the source dataset did not show a sufficient level of consensus. To demonstrate the quality of the methodology, we show how the model performs on a sequence taken from the test set (EG11982): Figure 5 displays our predicted speed in comparison with the level assigned by the consensus pool.

Finally, we illustrate how our model behaves on the unlabeled ORFs. In Figure 6, we demonstrate the speed prediction on ORF "azoR" as an example. All the predictions on the other *E. coli* ORFs are available in the Supplementary Materials, both as plots and as source data in text format.

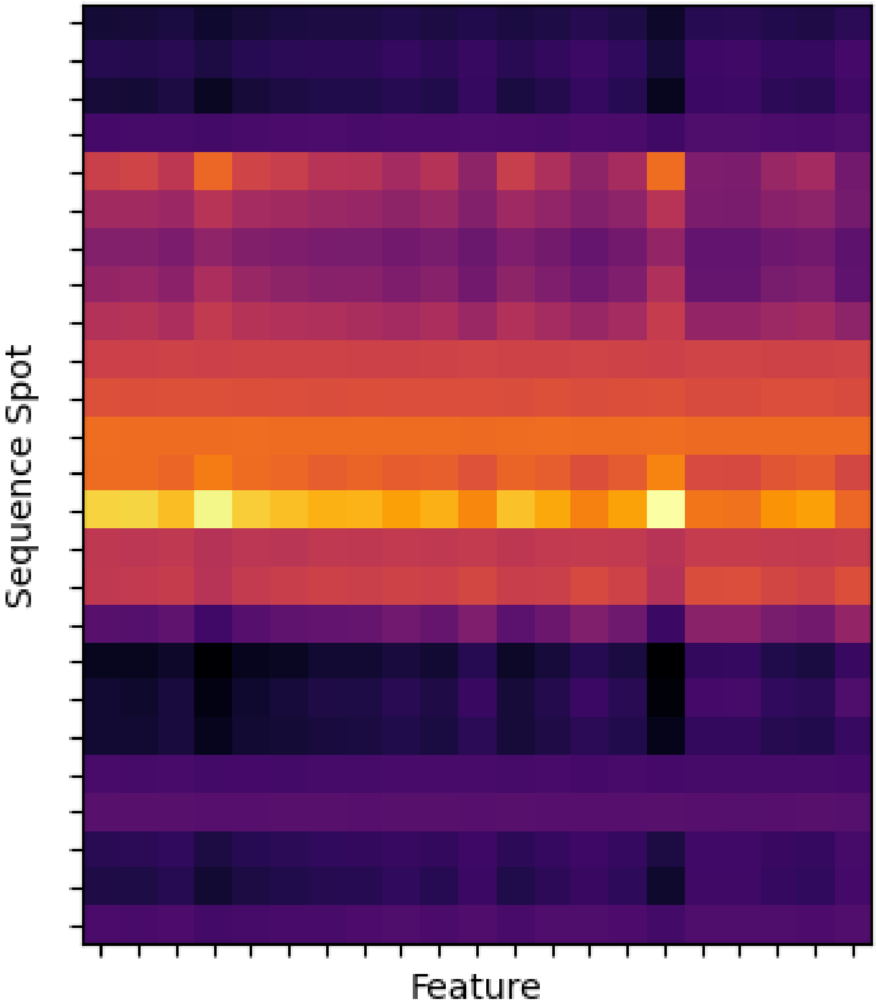

Figure 3. Attention heat-map obtained with the amino acid encoding and a context length of 12.

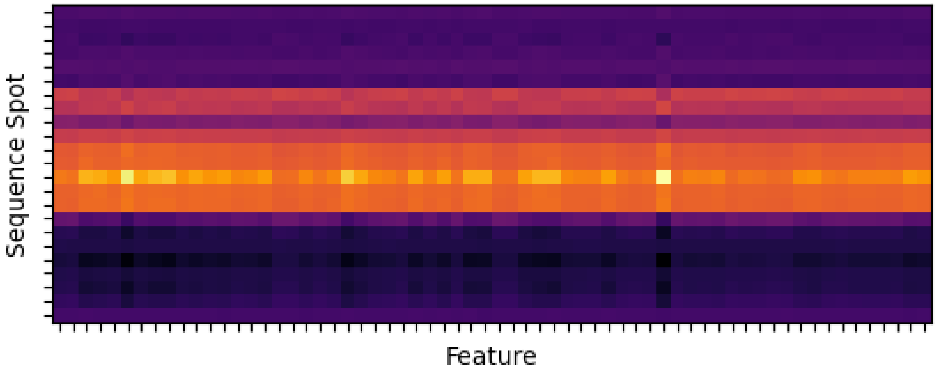

Figure 4. Attention heat-map obtained with the spread codon encoding and a context length of 11.

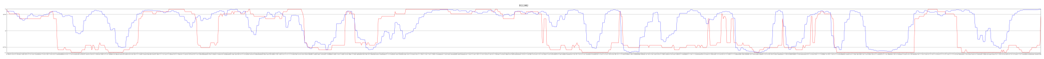

Figure 5. Prediction realized with the spread codon encoding and a context length of 11 on test sequence EG11982. The prediction is shown in blue, while the target calculated from the consensus pool is shown in red for comparison.

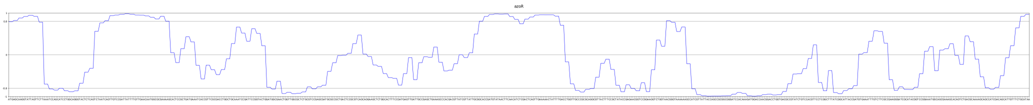

Figure 6. Prediction realized with the spread codon encoding and a context length of 11 on ORF "azoR". The prediction is shown in blue.

4. Conclusions

In this paper, we described a methodology for the point-wise ribosome translation speed prediction of mRNA sequences. Four machine learning models were applied to the task and compared: 1D Convolutional Neural Networks, Long Short-Term Memories, Graph Neural Networks, and hybrid Graph Neural Networks with LSTM aggregation. The models were trained, validated, and tested on a dataset of 49 high-confidence sequences obtained from a consensus pool of nine source datasets of Ribo-Seq profile lab measurements on the complete set of *E. coli* ORFs. In particular, the objective was to produce a point-by-point estimate of the translation speed on every sequence element. The problem was presented to the model as a classification one, with a + 1 target for fast points, a − 1 target for slow points, and a 0 target for points with a speed close to the average. The model could also be exploited as an estimator of the speed, as it produced a prediction in a continuous domain through a sigmoid output unit, which could take any value from −1 to +1. The dataset was presented to each model in four different encodings, accounting for nucleotides, amino acids, codons (as triples of nucleotides), and split codons (64-bit one-hot encodings). We carried out a grid search, obtaining the optimal configuration of every model, which demonstrated that networks with a relatively small number of parameters are capable of processing the sequences with very good results. Moreover, a comparison between the different architectures showed that LSTMs are the best models for this task, thanks to their natural way of processing sequential inputs. We subsequently employed LSTMs to carry out experiments on our sequences with contexts of different lengths. The length of the context was the width of the sliding window we took into account when making predictions on every sequence point. These experiments allowed us to determine that the best context length is 11 codons or 12 amino acids, depending on the encoding. Future work will focus on better understanding the mechanism that brings the networks to take into account this span. An LSTM model with attention layers allows measuring the importance of each context element and each amino acid, codon, or nucleotide. Finally, further experiments on the lengths of asymmetric contexts would allow us to more precisely determine the region that actually interacts with the ribosome during translation, thus contributing to its translation speed.

Supplementary Materials: The following supporting information can be downloaded at: www.mdpi.com/xxx/s1, Translation speed predicted on all the *E. coli* ORFs.

Author Contributions: Conceptualization, P.B., N.P. and M.B; methodology, P.B.; software, N.P. and P.B.; validation, P.A. and G.G.; formal analysis, V.L. and C.G.; investigation, M.B., P.B. and N.P.; resources, N.P. and P.B.; data curation, N.P. and P.B.; writing—original draft preparation, P.B.; writing—review and editing, all the authors; visualization, N.P.; supervision, P.B. and M.B.; project administration, M.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The dataset and the translation speed predictions of *E. coli* ORFs are available in the Supplementary Materials.

Conflicts of Interest: The authors declare no conflict of interest

Abbreviations

The following abbreviations are used in this manuscript:

1D–CNN	One Dimensional Convolutional Neural Network
CNN	Convolutional Neural Network
E. coli	*Escherichia coli*
GEO	Gene Expression Omnibus
GNN	Graph Neural Network
LSTM	Long Short-Term Memory
MLP	Multi–Layer Perceptron
mRNA	messenger Ribo–Nucleic Acid
ORF	Open Reading Frame
Ribo-Seq	Ribosome Sequencing profiling
RNA	Ribo–Nucleic Acid
RNN	Recurrent Neural Network

References

1. Cao, R. mTOR signaling, translational control, and the circadian clock. *Front. Genet.* **2018**, *9*, 367. [CrossRef] [PubMed]
2. Charneski, C.A.; Hurst, L.D. Positively charged residues are the major determinants of ribosomal velocity. *PLoS Biol.* **2013**, *11*, e1001508. [CrossRef] [PubMed]
3. Archer, S.K.; Shirokikh, N.E.; Beilharz, T.H.; Preiss, T. Dynamics of ribosome scanning and recycling revealed by translation complex profiling. *Nature* **2016**, *535*, 570–574. [CrossRef] [PubMed]
4. Edgar, R.; Domrachev, M.; Lash, A.E. Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. *Nucleic Acids Res.* **2002**, *30*, 207–210. [CrossRef] [PubMed]
5. Valleriani, A.; Chiarugi, D. A workbench for the translational control of gene expression. *bioRxiv* **2020**. . [CrossRef]
6. Woolstenhulme, C.J.; Guydosh, N.R.; Green, R.; Buskirk, A.R. High-precision analysis of translational pausing by ribosome profiling in bacteria lacking EFP. *Cell Rep.* **2015**, *11*, 13–21. [CrossRef]
7. Morgan, G.J.; Burkhardt, D.H.; Kelly, J.W.; Powers, E.T. Translation efficiency is maintained at elevated temperature in *Escherichia coli*. *J. Biol. Chem.* **2018**, *293*, 777–793. [CrossRef]
8. Mohammad, F.; Woolstenhulme, C.J.; Green, R.; Buskirk, A.R. Clarifying the translational pausing landscape in bacteria by ribosome profiling. *Cell Rep.* **2016**, *14*, 686–694. [CrossRef]
9. Li, G.W.; Burkhardt, D.; Gross, C.; Weissman, J.S. Quantifying absolute protein synthesis rates reveals principles underlying allocation of cellular resources. *Cell* **2014**, *157*, 624–635. [CrossRef]
10. Subramaniam, A.R.; Zid, B.M.; O'Shea, E.K. An integrated approach reveals regulatory controls on bacterial translation elongation. *Cell* **2014**, *159*, 1200–1211. [CrossRef]
11. Guo, M.S.; Updegrove, T.B.; Gogol, E.B.; Shabalina, S.A.; Gross, C.A.; Storz, G. MicL, a new σE-dependent sRNA, combats envelope stress by repressing synthesis of Lpp, the major outer membrane lipoprotein. *Genes Dev.* **2014**, *28*, 1620–1634. [CrossRef]
12. Burkhardt, D.H.; Rouskin, S.; Zhang, Y.; Li, G.W.; Weissman, J.S.; Gross, C.A. Operon mRNAs are organized into ORF-centric structures that predict translation efficiency. *Elife* **2017**, *6*, e22037. [CrossRef] [PubMed]

13. Li, G.W.; Oh, E.; Weissman, J.S. The anti-Shine–Dalgarno sequence drives translational pausing and codon choice in bacteria. *Nature* **2012**, *484*, 538–541. [CrossRef] [PubMed]
14. Baggett, N.E.; Zhang, Y.; Gross, C.A. Global analysis of translation termination in *E. coli*. *PLoS Genet.* **2017**, *13*, e1006676. [CrossRef] [PubMed]
15. Giacomini, G.; Graziani, C.; Lachi, V.; Bongini, P.; Pancino, N.; Bianchini, M.; Chiarugi, D.; Valleriani, A.; Andreini, P. A Neural Network Approach for the Analysis of Reproducible Ribo–Seq Profiles. *Algorithms* **2022**, *15*, 274. [CrossRef]
16. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
17. LeCun, Y.; Bengio, Y. Convolutional networks for images, speech, and time series. *Handb. Brain Theory Neural Netw.* **1995**, *3361*, 1995.
18. Scarselli, F.; Gori, M.; Tsoi, A.C.; Hagenbuchner, M.; Monfardini, G. The graph neural network model. *IEEE Trans. Neural Netw.* **2008**, *20*, 61–80. [CrossRef]
19. Gardin, J.; Yeasmin, R.; Yurovsky, A.; Cai, Y.; Skiena, S.; Futcher, B. Measurement of average decoding rates of the 61 sense codons in vivo. *Elife* **2014**, *3*, e03735. [CrossRef]
20. Chollet, F. Keras v1.0. 2015. Available online: https://keras.io (accessed on 28 January 2024).
21. Pancino, N.; Bongini, P.; Scarselli, F.; Bianchini, M. GNNkeras: A Keras-based library for Graph Neural Networks and homogeneous and heterogeneous graph processing. *SoftwareX* **2022**, *18*, 101061. [CrossRef]
22. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**. arXiv:1412.6980.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.